

Claims

We claim:

*Sub
a1*

1. A computer-implemented method for invoking execution of a graphical program, the method comprising:

calling an entry point function of a shared library, wherein the entry point function is associated with the graphical program;

the graphical program executing in response to said calling the entry point function.

10

2. The method of claim 1, further comprising:

creating the shared library including the entry point function associated with the graphical program.

15

3. The method of claim 1, further comprising:

the entry point function invoking the graphical program in response to said calling the entry point function;

the graphical program executing in response to said invoking the graphical program.

20

*Sub
a2*

4. The method of claim 3,

wherein the graphical program comprises executable code included in the shared library.

25

5. The method of claim 3,

wherein said calling the entry point function comprises passing one or more parameters to the entry point function;

wherein said entry point function invoking the graphical program comprises the entry point function passing the one or more parameters to the graphical program.

CONFIDENTIAL

CONFIDENTIAL

6. The method of claim 3, wherein the graphical program produces one or more output values, the method further comprising:

- 5 the graphical program returning the one or more output values to the entry point function;
- the entry point function returning the one or more output values.

7. The method of claim 3, wherein said calling the entry point function comprises passing a parameter to the entry point function, the method further comprising:

- 10 the entry point function transforming the parameter into a format expected by the graphical program;
- the entry point function passing the transformed parameter to the graphical program.

8. The method of claim 3, wherein said calling the entry point function comprises passing a parameter to the entry point function, the method further comprising:

- 15 the entry point function copying the parameter into a location expected by the graphical program.

9. The method of claim 1, wherein said calling the entry point function is performed by a particular thread;

wherein the graphical program executes within the context of the particular thread.

10. The method of claim 2, wherein the graphical program has an associated input, the method further comprising:

- 25 specifying a functional interface for the graphical program, wherein the functional interface maps the graphical program input to an input parameter;

wherein said creating the shared library comprises creating an entry point function in accordance with the functional interface.

11. The method of claim 2, wherein the graphical program has an associated
5 output, the method further comprising:

specifying a functional interface for the graphical program, wherein the functional interface maps the graphical program output to an output parameter;

wherein said creating the shared library comprises creating an entry point function in accordance with the functional interface.

10

12. The method of claim 2,

wherein said creating the shared library comprises excluding a portion of the graphical program that is not necessary for execution.

15

13. The method of claim 12,

wherein the portion of the graphical program that is not necessary for execution comprises information from the group consisting of:

user interface display information and block diagram information.

20

14. The method of claim 1,

wherein the shared library is one of:

a Windows DLL, a Unix shared library, and a Macintosh code fragment.

15. The method of claim 1,

25

wherein said calling the entry point function of the shared library is performed by a program created using a text-based programming language.

16. A computer-implemented method for creating a shared library including a graphical program, the method comprising:

CONFIDENTIAL

AZ

CONFIDENTIAL

5 selecting a graphical program in response to user input;
specifying a functional interface for the graphical program;
creating a shared library comprising the graphical program, wherein the shared
library includes a function created according to the functional interface specified for the
graphical program.

17. The method of claim 16,
wherein the graphical program has associated inputs and outputs;
wherein said specifying a functional interface for the graphical program comprises
10 a user specifying a mapping of the associated inputs and outputs to parameters in a
function declaration.

18. The method of claim 16,
wherein the graphical program has associated inputs and outputs;
15 wherein said specifying a functional interface for the graphical program comprises
automatically mapping the associated inputs and outputs to parameters in a function
declaration.

19. The method of claim 16,
20 wherein, in response to being invoked by a program, the shared library function
created according to the functional interface specified for the graphical program is
operable to invoke the graphical program.

20. A system comprising:
25 a computer including a CPU and memory;
a shared library stored in the memory of the computer, wherein the shared library
includes an entry point function associated with a graphical program;
wherein, in response to a program calling the entry point function, the entry point
function is operable to cause the graphical program to execute.

21. The system of claim 20,
wherein said entry point function causing the graphical program to execute
comprises the entry point function invoking the graphical program.

5
*Sub
City*
22. The system of claim 20,
wherein the graphical program comprises executable code included in the shared
library.

10
23. The system of claim 21,
wherein said calling the entry point function comprises passing one or more
parameters to the entry point function;
wherein said entry point function invoking the graphical program comprises the
entry point function passing the one or more parameters to the graphical program.

15
24. The system of claim 21,
wherein the graphical program produces one or more output values;
wherein the graphical program returns the one or more output values to the entry
point function;
20
wherein the entry point function returns the one or more output values.

25
25. The system of claim 21,
wherein said calling the entry point function comprises passing a parameter to the
entry point function;
wherein the entry point function transforms the parameter into a format expected
by the graphical program;
wherein the entry point function passes the transformed parameter to the graphical
program.

CONFIDENTIAL

AM

26. The system of claim 21,
wherein said calling the entry point function comprises passing a parameter to the
entry point function;
wherein the entry point function copies the parameter into a location expected by
5 the graphical program.

27. The system of claim 20,
wherein said calling the entry point function is performed by a particular thread;
wherein the graphical program executes within the context of the particular
10 thread.

28. The system of claim 20,
wherein the shared library is one of:
a Windows DLL, a Unix shared library, and a Macintosh code fragment.
15

29. The system of claim 20,
wherein said calling the entry point function of the shared library is performed by
a program created using a text-based programming language.

30. A computer-implemented system for creating a shared library including a
graphical program, the system comprising:
a computer including a CPU and memory;
a graphical programming system stored in the memory;
wherein the graphical programming system is operable to:
25 select a graphical program in response to user input;
specify a functional interface for the graphical program;
create a shared library comprising the graphical program, wherein the
shared library includes a function created according to the functional interface specified
for the graphical program.

CONFIDENTIAL

5

10

15

20

25

31. The system of claim 30,
wherein the graphical program has associated inputs and outputs;
wherein said specifying a functional interface for the graphical program comprises
a user specifying a mapping of the associated inputs and outputs to parameters in a
function declaration.

32. The system of claim 30,
wherein the graphical program has associated inputs and outputs;
wherein said specifying a functional interface for the graphical program comprises
automatically mapping the associated inputs and outputs to parameters in a function
declaration.

33. The system of claim 30,
wherein, in response to being invoked by a program, the shared library function
created according to the functional interface specified for the graphical program is
operable to invoke the graphical program.

34. A memory medium comprising program instructions operable to:
call an entry point function of a shared library, wherein the entry point function is
associated with a graphical program;
execute the graphical program in response to said calling the entry point function.

35. The memory medium of claim 34, further comprising program instructions
operable to:
create the shared library including the entry point function associated with the
graphical program.

36. The memory medium of claim 34, further comprising program instructions operable to:

invoke the graphical program in response to said calling the entry point function;
wherein said invoking the graphical program comprises the graphical program
5 executing.

Sub
44
37. The memory medium of claim 36,
wherein the graphical program comprises executable code included in the shared
library.

10

38. The memory medium of claim 36,
wherein said calling the entry point function comprises passing one or more
parameters to the entry point function;

wherein said invoking the graphical program comprises the entry point function
15 passing the one or more parameters to the graphical program.

39. The memory medium of claim 36,
wherein the graphical program produces one or more output values
wherein the graphical program returns the one or more output values to the entry
20 point function;

wherein the entry point function returns the one or more output values.

40. The memory medium of claim 36,
wherein said calling the entry point function comprises passing a parameter to the
25 entry point function;

wherein the entry point function transforms the parameter into a format expected
by the graphical program;

wherein the entry point function passes the transformed parameter to the graphical
program.

CONFIDENTIAL

CONFIDENTIAL

41. The memory medium of claim 36,
wherein said calling the entry point function comprises passing a parameter to the
entry point function;

5 wherein the entry point function copies the parameter into a location expected by
the graphical program.

42. The memory medium of claim 34,
wherein said calling the entry point function is performed by a particular thread;
10 wherein the graphical program executes within the context of the particular
thread.

43. The memory medium of claim 35, wherein the graphical program has an
associated input, the memory medium further comprising program instructions operable
15 to:

specify a functional interface for the graphical program, wherein the functional
interface maps the graphical program input to an input parameter;

wherein said creating the shared library comprises creating an entry point function
in accordance with the functional interface.

44. The memory medium of claim 35, wherein the graphical program has an
associated output, the memory medium further comprising program instructions operable
20 to:

specify a functional interface for the graphical program, wherein the functional
25 interface maps the graphical program output to an output parameter;

wherein said creating the shared library comprises creating an entry point function
in accordance with the functional interface.

45. The memory medium of claim 35,

wherein said creating the shared library comprises excluding a portion of the graphical program that is not necessary for execution.

At 5

46. The memory medium of claim 45,
wherein the portion of the graphical program that is not necessary for execution comprises information from the group consisting of:
user interface display information and block diagram information.

CONFIDENTIAL